DOI: 10.1111/cgf.12703
Eurographics Symposium on Geometry Processing 2015
Mirela Ben-Chen and Ligang Liu
(Guest Editors)

Volume 34 (2015), Number 5

LeSSS: Learned Shared Semantic Spaces for Relating Multi-Modal Representations of 3D Shapes

Robert Herzog[1], Daniel Mewes[1], Michael Wand[2], Leonidas Guibas[3], Hans-Peter Seidel[1]

[1]Computer Graphics Department, MPI Informatik, [2]Institute of Computer Science, University of Mainz,
[3]Computer Science Department, Stanford University

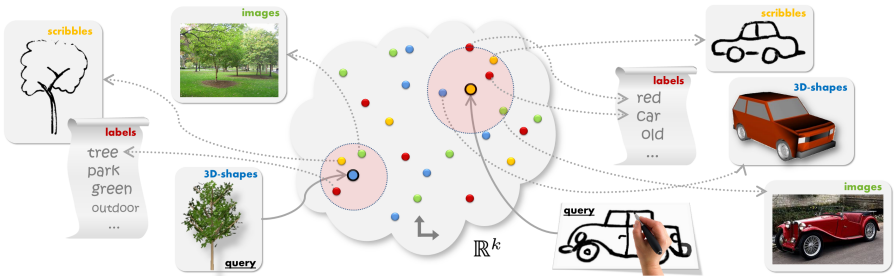


Figure 1: *Learning a shared semantic space from multiple annotated databases enables to retrieve and classify objects across different modalities, including 3D-shapes, text, images, and user scribbles.*

Abstract

In this paper, we propose a new method for structuring multi-modal representations of shapes according to semantic relations. We learn a metric that links semantically similar objects represented in different modalities. First, 3D-shapes are associated with textual labels by learning how textual attributes are related to the observed geometry. Correlations between similar labels are captured by simultaneously embedding labels and shape descriptors into a common latent space in which an inner product corresponds to similarity. The mapping is learned robustly by optimizing a rank-based loss function under a sparseness prior for the spectrum of the matrix of all classifiers. Second, we extend this framework towards relating multi-modal representations of the geometric objects. The key idea is that weak cues from shared human labels are sufficient to obtain a consistent embedding of related objects even though their representations are not directly comparable. We evaluate our method against common base-line approaches, investigate the influence of different geometric descriptors, and demonstrate a prototypical multi-modal browser that relates 3D-objects with text, photographs, and 2D line sketches.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.5]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations; Artificial Intelligence [I.2.10]: Vision and Scene Understanding—Shape Image Processing and Computer Vision [I.4.8]: Scene Analysis—Object recognition

Keywords: multi-modal learning, object recognition, collaborative filtering, 3D-shape descriptors, semantic correspondences, object retrieval

1. Introduction

As digital sensors have become ubiquitous and the internet gives us access to enormous quantities of information, structuring and understanding data has become one of the big challenges of modern computer science. Computer graphics is no exception: Within the past decade, data-driven methods have become a central research focus. A lot of effort has recently been made [SMKF04, ATC*05, MGGP06, RES*06, LG07, KHS10, vKZHCO11, WBU11, KFLCO13, HSS*13, ARS13, HFL14] to structure large data bases of images and

3D-models in order to extract information for the analysis and synthesis of shapes, images, and other media.

Our paper addresses the problem of relating geometric shapes to each other and to alternative representations (photos, 2D sketches, textual labels) of themselves. To this end, we learn a common similarity metric for multi-modal representations, linked by weak human labeling such as semantic tags in data bases. We proceed in two conceptual steps:

First, we consider the problem of learning to relate 3D shapes with each other based on weak textual labels. Weak labeling means that objects have been tagged with lists of keywords (for example, "car", "sports car", "fast", "old", "vintage") that capture human semantic categories. Labels might be redundant (e.g., synonyms, different languages), noisy (e.g., imprecise, overly generic, mistakes), and partially related (such as "cars" and "sport cars"). Our objective is to learn the similarity of shapes to each other (and thereby also to the associated labels).

We employ recent ideas from image understanding and recommendation systems [RS05, LF08, LFEF09, WBU11] to address this multi-class learning problem. Instead of learning separate classifiers for each class label, we assume that classifiers can be linearly combined from a more compact, lower-dimensional space of basis classifiers. Implicitly, this retrieves similarities between related labels and shapes and extracts more information from the training data. This can also be viewed as an instance of a *collaborative filtering* technique, as used originally in recommendation systems [SK09] to discover hidden correlations in sparsely sampled data. Technically, labels and shape descriptors are modeled as points that are embedded into a common space with the objective of nearest neighbors representing the most similar objects. Embedding is performed using a linear projection matrix, and a penalty on its trace-norm encourages information sharing.

In a second step, we extend this framework towards learning similarities in multi-modal object representations (3D geometry, photographs, and 2D sketches, all annotated with sparse and weak textual labels). Again, the different types of objects and labels are embedded as points in a common space with the objective of agreeing on the available labels. The resulting embedding provides a metric to relate objects of different modalities, which can be used for browsing, querying, and navigating the dataset.

The rationale for this design is that most data bases naturally come with weak (noisy/unreliable) label sets, but these are most often not accurate enough to directly navigate the data. Our approach merely uses this noisy information to calibrate similarity measures of the geometric and visual data itself. Such data provides richer information than the label sets, but misses semantic information. By combining both sources of information we can structure the data even though a direct comparison of the different modalities would be very difficult. Further, after learning from weakly labeled data, new, unlabeled objects can be understood using the learned geometry/appearance-based classifiers.

We evaluate our approach on three different benchmark datasets (Princeton shape benchmark, a collection of Trimble 3D Warehouse models, and the "LabelMe" dataset [RTMF07]), and study the influence of a large selection of geometric descriptors on the obtained performance. Furthermore, we show that the proposed learning-based approach is competitive with a carefully hand-crafted design for explicit cross-modal matching [ERB*12], yielding an interesting, generic alternative for associating geometric data with non-geometric attributes and representations. We also provide a prototypical cross-modal object-retrieval application for browsing data bases.

In summary, our paper makes the following contributions: First, we propose and evaluate a scalable system for jointly learning annotations of 3D-shapes with multiple correlated labels. This part transfers ideas from related disciplines, which, to the best of our knowledge, have not been applied before in the context of 3D geometry. Second, we generalize the uni-modal approach to multi-modal structuring of data, specifically *text labels, images, 2D sketches, and 3D-shapes*, providing a simple and generic way of associating geometry with various other representations.

2. Related Work

Relating shapes has recently received a lot of interest. Direct shape matching [vKZHCO11] is well understood for classes of shapes that are related by fixed groups of mappings, such as extrinsic or intrinsic isometries. In contrast, relating shapes of similar semantics or functionality is much more difficult and requires more invariant shape representations [TV08]. Similar to several previous systems, our method employs bags-of-features [LSP06, MGGP06, LG07, BBGO11, LBBC14] to characterize shapes globally by building histograms of local descriptors. As local descriptor, we use an adaptation of the very successful histograms of oriented gradients [Low04, DT05] to 3D-shapes, where image gradients are replaced by geometric crease lines [MFK*10, SKVS13, KBWS13, STS14, RMHM14, SX14]. The descriptor design can easily be varied; we evaluate performance on a larger set of alternative descriptors.

"Semantic" similarity is subjective and often application-dependent; therefore, machine-learning is required to adapt the notion of similarity to the user's intent. In our case, we assume that large quantities of weakly labeled data is available, as typically provided by online repositories. Here, labels are not mutually exclusive but semantically correlated, and our goal is to utilize this to improve performance (i.e., not using simple one-vs-all classifiers [FCH*08]). In 3D-shape analysis, this has not yet been attempted, but a

number of methods have been proposed for image classification [FEHF09, FEH10]. One approach is collaborative filtering [SK09]: By assuming that the classifiers can be represented within a dimensionality-reduced subspace, sharing and utilizing of semantic similarity is encouraged; this can be formulated implicitly by a sparseness prior on the singular value spectrum of the matrix of all classifiers [RS05,LF08,LFEF09]. We base our method on an extension of the method of Weston et al. [WBU11] that combines this idea with stochastic large-margin ranking [CB04]. We were also inspired by the extension described in [WBH11] that, in the context of music retrieval, structures different types of labels (artist, genre, etc.). Different from this, our method uses only one class of labels but links different data modalities.

Kleiman et al. [KFLCO13] propose a method for computing a dynamic shape layout for browsing shape repositories. Focusing on exploration, it does not aim at providing a globally consistent metric. Further, their approach is single modal and based on direct geometric similarity. Huang et al. [HSS*13] introduce quartet analysis from bio-informatics for structuring data that cannot be described by a global metric but rely on local similarity.

Sketch-based retrieval is a specific example of cross-modal data matching: Eitz et al. [ERB*12] use perceptually weighted light field descriptors and contour rendering to enable a direct comparison of geometry and image data. Learning can improve performance [ST14]; our approach is different as we do not learn a direct regression function from sketches to geometry but rather aim at a label-guided co-embedding of multiple modalities, which offers more flexibility in integrating various representations without relating all pairs of types and permits utilizing label correlation information. Aubry et al. [ARS13] propose a method for aligning paintings with geometry by learning the most discriminative features that can be matched reliably across modalities. This idea is orthogonal to our embedding approach. The same holds for utilizing additional cues, such as context [XMZ*14].

3. Correlated Discriminative Features

3.1. Feature Detection

As usual, we first transform local pieces of geometry into a more invariant representation by feature vectors. We examine a variety of choices to assess the influence on our framework. We use the established spherical harmonics [KFR03] and light-field descriptors [CXPYTO03, ERB*12]. Further, we add two implementations of the more recent class of descriptors based on histograms of differential properties [ZBVH09, SKVS13, KBWS13, RMHM14, STS14, SX14]. These have shown excellent invariance properties at still high specificity by focusing on salient creases/gradients and using spatial and orientational histograms to achieve local

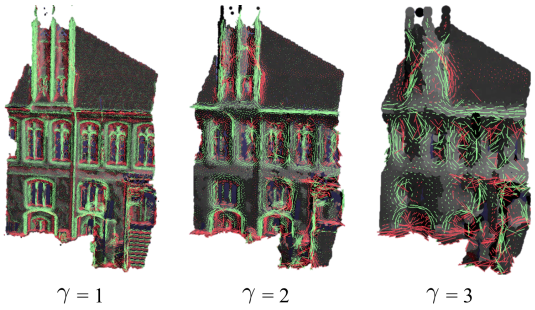


Figure 2: *Multi-scale feature detection for 3D-shapes using "histograms of oriented curvature (HOC)". We determine curvature vectors at multiple levels of detail as analogs to image gradients. (signed curvature vectors – green: positive curvature, red: negative).*

deformation invariance. Below, we further detail our implementation for reproducibility.

Histograms of Oriented Curvature (HOC): This descriptor is based on crease lines [KBWS13]: We assume that we are given a point cloud with oriented normals and sample spacing ε (triangle meshes are converted into point clouds via Poisson-disc sampling). We then approximate a smooth surface via quadratic moving-least-squares and estimate a local, average tangent frame (with normal $\mathbf{n}$) and the average curvature tensor [CP05], from which we obtain a direction of maximum principal curvature $\mathbf{t}_1$ and its magnitude κ_1. We convert this information into *curvature vectors* $\kappa_1(\mathbf{n} \times \mathbf{t}_1)$ that serve as an analog of 2D-image gradients for 3D-surfaces. We now project the curvature vectors into an average tangent plane and collect orientation statistics with 8 orientation bins and 4×4 spatial bins (see Fig. 3), each with an edge length of 5ε as described in detail in [KBWS13]. We ignore the direction of the curvature vectors (i.e., employ binning modulo 180°) because normal directions are usually unreliable in generic 3D data bases. For capturing information in different frequency bands and for obtaining scale-invariant matching, we employ a multi-scale framework (Fig. 2). We downsample the point cloud repeatedly (using a Gaussian window for anti-aliasing), enlarging ε by a constant factor of γ (in practice, we use 5 levels, $\gamma = 1.5$). The result is a set of scale-annotated descriptors. Note that our HOC descriptor can be regarded as a specialized version of the MeshHOG descriptor [ZBVH09].

Histogram of Oriented Normals (HON): We obtain a second variant by replacing curvature vectors with surface normals [STS14], keeping the rest of the descriptor design unchanged. HON descriptors can distinguish flat surfaces from empty space. Edges and corners, on the other hand, obtain less weight. HON descriptors are targeted towards shapes with coarse, possibly smooth feature structures, while HOC descriptors appear more suitable for relief-like structures (such as architecture/façades).

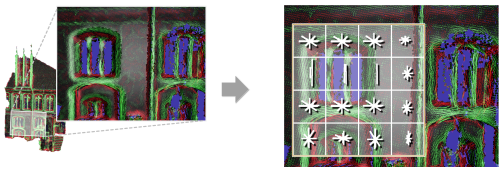


Figure 3: *HOC descriptors accumulate curvature orientation in the tangent plane of a surface, employing coarse spatial (4×4) and orientational (8 fold modulo 180°) binning for invariance.*

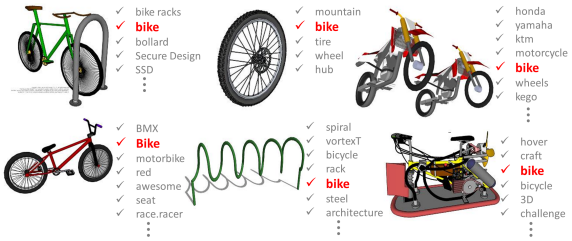


Figure 4: *Example 3D data with noisy and correlated labels showing a subset for label "bike" from Trimble 3D Warehouse™.*

3.2. Correlated Classifiers and Attributes

We turn our attention towards the problem of learning an association between multiple weak and correlated textual labels and 3D geometry. This is done by optimizing the placement of labels and objects in a joint, latent Euclidean space such that the dot product of feature vectors of similar objects is maximized. We use a ranking objective that aims at making pairs with a high dot product consistent. As regularizer, we impose a soft constraint on the dimension of the joint space, encouraging sharing of information between different classes of labels and avoiding overfitting.

3.2.1. Input and Preprocessing

Training data: First of all, we assume that we are given a *training* set of objects with comprehensive annotations, as shown in Figure 4. We validate the results on a *test* set, where annotations are hidden to the algorithm. For now, we restrict ourselves to 3D-meshes and postpone the discussion of other modalities to Section 4. In our experiments, we use 3D-meshes from a large collection of objects from Trimble 3D Warehouse™ (abbr. "WH") as well as the smaller set of labeled shapes from the Princeton Shape Benchmark [SMKF04] ("PSB"). Annotations are taken from the labels provided by the data base. In case of WH the annotations are numerous and rather noisy and we initially filter rare labels if they show up less than 10 times in the entire dataset. We also assume that models have a consistent upward orientation. Abstractly, we denote the 3D-models as n separate 3D point sets $\mathcal{S}_1, \ldots, \mathcal{S}_n \subset \mathbb{R}^3$.

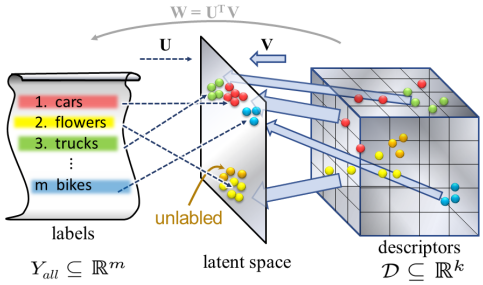


Figure 5: *The key idea of the employed collaborative multi-task learning approach is to compute an embedding of labels (left) and descriptors (right) into a common latent space. The inner product in the latent space reflects semantic similarity.*

Annotations: The training data is annotated with a total of m labels $\mathbf{y}_j \in Y_{all}$ (the terms "label" and "annotation" are used synonymously). We denote the list of annotations of sample $\mathcal{S}_i$ by Y_i. To simplify the notation later on, we represent each single label $\mathbf{y}_j$ by a binary vector, i.e., $\mathbf{y}_j \in \mathbb{R}^m$, where the j^{th} entry equals 1 and the rest is set to zero.

Descriptors: For each 3D-model $\mathcal{S}_i$, we compute our dense HOC-descriptors at different scales by first subsampling $\mathcal{S}_i$ with sample spacing 2.5ε (which is the Nyquist-frequency matching the spatial binning of the histograms). At each sample point, we center our HOC-descriptor, using the average surface normal and the fixed upward orientation to compute a local reference frame that fixes the rotational alignment. We denote the set of all descriptors by $\mathcal{D}$ and that of a given training sample by $\mathcal{D}_i$.

Bag-of-features: We convert the sets $\mathcal{D}_i$ into bags-of-features by the standard procedure of performing k-means clustering (with $k \approx 500$) and building histograms of nearest neighbors to the centroids. Additionally, we weight the clusters by the inverse of their sizes following the *term frequency-inverse document frequency* principle. We denote the resulting k-dimensional frequency vectors by $\mathbf{x}_i$ where $\|\mathbf{x}_i\|_2 = 1$.

3.2.2. Learning Algorithm

We learn m linear classifiers $\mathbf{w}_j : \mathbb{R}^k \to \mathbb{R}$ that detect whether a descriptor $\mathbf{x} \in \mathbb{R}^k$ is likely to carry the label $y_j, j = 1, .., m$. The detector acts like a linear support vector machine, i.e., given a descriptor $\mathbf{x}$, the score is just obtained by a scalar product with the classifier vector

$$score_j(\mathbf{x}) = \langle \mathbf{w}_j, \mathbf{x} \rangle \tag{1}$$

We use $\mathbf{W}$ to denote the $m \times k$ matrix of all classifiers, stacked row-wise. The main idea now is to interpret the mapping from classifiers to annotations as an embedding in a joint, latent space, in which the labels are positioned with their inner products reflecting their semantic similar-

ity. Correspondingly, descriptors are also mapped into the same space, and those that are indicative of a certain semantic label are embedded to similar directions of the space (see Fig. 5 and Fig. 6). Formally, we factor $\mathbf{W}$ into two parts:

$$\mathbf{W} = \mathbf{U}^T \mathbf{V} \tag{2}$$

The right handed matrix $\mathbf{V}$ maps descriptor vectors $\mathbf{x}$ into the latent space, and $\mathbf{U}$ maps labels $\mathbf{y}$ into the same space. With this, we have 2 linear mappings U and V that together constitute the classifier ensemble $W = U^T V$. This can be viewed as mapping labels $\mathbf{y}$ and features $\mathbf{x}$ to a unified semantic space

$$\mathbf{y}^* \leftarrow \mathbf{U} \cdot \mathbf{y} \quad \text{and} \quad \mathbf{x}^* \leftarrow \mathbf{V} \cdot \mathbf{x}, \tag{3}$$

respectively (in the following, superscript stars indicate vectors in the latent semantic space). In particular, we can measure the "semantic" similarity $f(\mathbf{x},\mathbf{y})$ between labels $\mathbf{y}$ and our features $\mathbf{x}$ using the similarity measure $d(\mathbf{y}^*,\mathbf{x}^*)$ in the latent space (see Fig. 5). In our paper, we define the similarity measure $d(\cdot,\cdot)$ in the latent space as $d(\mathbf{y}^*,\mathbf{x}^*) = \langle \mathbf{y}^*,\mathbf{x}^* \rangle$. This induces a function $f(\cdot,\cdot)$ in the original space:

$$f(\mathbf{x},\mathbf{y}) = d(\mathbf{x}^*,\mathbf{y}^*) \tag{4}$$

$$= \langle \mathbf{y}^*,\mathbf{x}^* \rangle = \mathbf{y}^T \mathbf{U}^T \mathbf{V}\mathbf{x} = \mathbf{y}^T \mathbf{W}\mathbf{x}. \tag{5}$$

Having setup a recognition model, we now address the problem of learning it from training data. We employ an objective function

$$E_{train}(\mathbf{W}) = E_D(\mathbf{W}) + E_R(\mathbf{W}) \tag{6}$$

that we seek to minimize in order to find optimal classifiers. The energy consists of two parts, a *loss function* $E_D(\mathbf{W})$ that penalizes bad alignment of the classification results with the available training data, and a *regularizer* $E_R(\mathbf{W})$ that defines the learning model and controls overfitting.

Regularization: The key ingredient is a regularization term that tries to keep the latent semantic space low-dimensional. Intuitively, we want the matrix $\mathbf{W}$ to have low rank, which forces the individual classifiers to share information. A straightforward choice would be to consider the squared Frobenius norm $\|\mathbf{W}\|_F^2$ of $\mathbf{W}$, which penalizes the squares of the singular values of $\mathbf{W}$, thereby spreading out the spectrum of $\mathbf{W}$ rather than encouraging low-rank solutions. A better choice is to penalize the L_1-norm of the singular values of $\mathbf{W}$ (trace norm), which aims at creating a sparse spectrum, preferring fewer larger singular values over many smaller ones. This is equivalent to constraining the squared Frobenius norm of the factors $\mathbf{U},\mathbf{V}$ [RS05]:

$$E_R(\mathbf{W}) = \frac{1}{2C}(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) \tag{7}$$

This *collaborative filtering* approach exploits the correlation in the label classification and enables to learn latent labels that have not been modeled explicitly in the feature descriptor $\mathbf{x}$. Such a didactic example is shown in Fig. 6 where colored fruit images were learned with 2D HOG features that have no notion of color in the descriptor. Nevertheless, due to the high correlation in the fruits and colors the multi-task learning was able to cluster colors and fruits in a meaningful way (e.g., "yellow", "banana").

Loss function: Assume that we are given l pairs of labeled training data $(\mathbf{x}_i,\mathbf{y}_i)_{i=1..l}$, where $\mathbf{x}_i$ denotes the descriptor and $\mathbf{y}_i$ a label for this descriptor that has been observed in the annotations of one of the example models. We now define a data loss function $E_D(\mathbf{W})$ that penalizes deviation from the training annotations. One option at this point is to use a simple square loss or, more effectively, the max-margin hinge loss (as in a support vector machine) to train a linear classifier $\mathbf{W}$ [RS05, LF08, LFEF09]. However, it has been shown [WBU11] that learning a rank-optimized linear classifier that favors a high precision at a few top-ranked data samples rather than optimizing the average precision over the entire training dataset outperforms these baseline classifiers for large datasets with many labels. We use this idea for multi-class learning with ranking and consider sets of labels Y per data sample, i.e., $(\mathbf{x}_i,Y_i)_{i=1..l}$. However, to simplify notation in the following derivations we will restrict ourselves to only one label $\mathbf{y}_i \in Y_i$. The energy we want to minimize over all training data using the rank-loss is given by:

$$E_D(\mathbf{W}) = \frac{1}{l}\sum_{i=1}^{l} \mathrm{L}_r(\mathrm{rank}(\mathbf{x}_i,\mathbf{y}_i)), \tag{8}$$

where $\mathrm{rank} \in [0,..,m)$ gives the position of the similarity score $f(\mathbf{x}_i,\mathbf{y}_i)$ corresponding to ground-truth label $\mathbf{y}_i$ in the list of all m scores $(\mathbf{W}\mathbf{x}_i)$ sorted in descending order. Intuitively, we measure whether the correct labels of the training data show up in the top detections when ranking them by their detector score. Formally, we can write:

$$\mathrm{rank}(\mathbf{x},\mathbf{y}) = \sum_{\mathbf{y}_k \neq \mathbf{y}} I\left(f(\mathbf{x},\mathbf{y}_k) > f(\mathbf{x},\mathbf{y})\right), \tag{9}$$

where I is the indicator function. L_r transforms this rank non-linearly into a loss such that precision is optimized at the top ranks:

$$\mathrm{L}_r(k) := \int_1^{k+1} \frac{1}{x} dx = \ln(k+1) \tag{10}$$

Importantly, errors among the first entries of the ranked list are penalized relatively high compared to ranking errors further on in the list, which makes it robust against outliers.

In [WBU11] Eq. (8) was changed to a continuous max-margin formulation where the indicator function in Eq. (9) is replaced by the continuous hinge-loss $h(\mathbf{x},\mathbf{y},\mathbf{y}_k) = \max(0, 1 - f(\mathbf{x},\mathbf{y}) + f(\mathbf{x},\mathbf{y}_k))$ resulting in

$$E_D^+(\mathbf{W}) = \frac{1}{l}\sum_{i=1}^{l} \mathrm{L}_r(\mathrm{rank}(\mathbf{x}_i,\mathbf{y}_i)) \sum_{\mathbf{y}_k \neq \mathbf{y}_i} \frac{h(\mathbf{x}_i,\mathbf{y}_i,\mathbf{y}_k)}{\mathrm{rank}(\mathbf{x}_i,\mathbf{y}_i)}. \tag{11}$$

Eq. (11) still contains non-differentiable terms and is inefficient to solve in a gradient based optimization scheme since for each sample $(\mathbf{x}_i,\mathbf{y}_i)$ we need to sum over all possible labels $\mathbf{y}_k \neq \mathbf{y}_i$. However, [WBU11] have shown how

to use an unbiased randomized sampling algorithm to efficiently approximate the function and overcome this deficiency. We randomly draw (with replacement) a label $\hat{\mathbf{y}} \neq \mathbf{y}$ with $h(\mathbf{x},\mathbf{y},\hat{\mathbf{y}}) > 0$, which has a probability $1/\text{rank}(\mathbf{x},\mathbf{y})$ or zero if $\text{rank}(\mathbf{x},\mathbf{y}) = 0$. The number of random trials $\hat{n}$ needed to find a violating label $\hat{\mathbf{y}}$ follows a geometric distribution where the success probability of each trial is $p = \frac{\text{rank}(\mathbf{x},\mathbf{y})}{m-1}$ with $m-1$ being the number of all labels excluding $\mathbf{y}$. Hence, the rank can be estimated as

$$\text{rank}(\mathbf{x},\mathbf{y}) \approx \frac{m-1}{\hat{n}}, \tag{12}$$

since the expectation for the number of Bernoulli trials $\hat{n}$ to find $\hat{\mathbf{y}}$ is $E[\hat{n}] = 1/p$. To this end, we can sample a data pair $(\mathbf{x},\mathbf{y})$ and a violating label $\hat{\mathbf{y}}$ and get an estimate for Eq. (11) with contribution

$$\mathrm{L}_r\left(\frac{m-1}{\hat{n}}\right) h(\mathbf{x},\mathbf{y},\hat{\mathbf{y}}). \tag{13}$$

Based on this loss-estimate, we minimize our objective $E_{train}(\mathbf{W})$ with stochastic gradient descent (SGD), which often yields a good generalization performance [BB08] and has been shown to be computational more efficient than traditional linear SVMs for large scale learning [WBU11, FGMR10, WdW10].

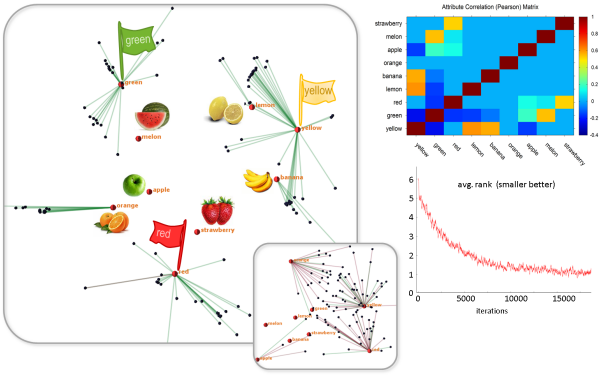


Figure 6: *Multi-task rank-loss training in a 2D latent space for a didactic dataset consisting of 100 images with 9 labels showing colorful fruits. Red dots represent labels, black dots the images mapped into this latent space. Lines link images to their top-ranked label (green: correctly labeled, purple: otherwise). The inset: the initial latent space; large image: the converged space after 15.000 SGD iterations – semantically related labels cluster naturally in this space; top right image: label correlation matrix, bottom right image: average rank-error after each SGD iteration.*

For performing SGD we need to compute a sub-gradient with respect to our model parameters $\mathbf{W} = \mathbf{U}^T\mathbf{V}$ for a single random sample $(\mathbf{x}_i,\mathbf{y}_i)$ with a contribution computed by Eq. (13). The sub-gradient with respect to U and V of the non-differentiable hinge-loss h is

$$\frac{\partial h}{\partial U}(\mathbf{x},\mathbf{y},\hat{\mathbf{y}}) = \begin{cases} (\hat{\mathbf{y}}-\mathbf{y})\cdot(\mathbf{V}\mathbf{x})^T & \text{if } h(\mathbf{x},\mathbf{y},\hat{\mathbf{y}}) > 0, \\ 0 & \text{else} \end{cases}$$
$$\frac{\partial h}{\partial V}(\mathbf{x},\mathbf{y},\hat{\mathbf{y}}) = \begin{cases} \mathbf{U}(\hat{\mathbf{y}}-\mathbf{y})\cdot\mathbf{x}^T & \text{if } h(\mathbf{x},\mathbf{y},\hat{\mathbf{y}}) > 0, \\ 0 & \text{else} \end{cases} \tag{14}$$

This approach is a margin perceptron, which has been shown to be equivalent to linear SVMs [CB04].

Finally, generalizing the previous derivations to multiple labels per sample is straight-forward and in summary each single SGD iteration consists of only a few simple steps:

- uniformly sample a training pair $(\mathbf{x}_i, Y_i)$
- sample a label $\mathbf{y}_i \in Y_i$
- sample with rejection a label $\hat{\mathbf{y}} \notin Y_i$ with $h(\mathbf{x}_i,\mathbf{y}_i,\hat{\mathbf{y}}) > 0$
- compute the rank-loss estimate $\mathrm{L}_r\left(\frac{m-|Y_i|}{\hat{n}}\right)$ (Eq. (10))
- compute the sub-gradient of $\nabla E_{train}(\mathbf{W})$ using Eq. (14) weighted by the rank-loss estimate
- perform gradient descent on $\mathbf{U}$ and $\mathbf{V}$ with fixed learning rate τ.

An example of the convergence results of our SGD learning algorithm is shown in Fig. 6 and in the accompanying video.

In practice, a single SGD iteration can still be costly and we sample the data in small random batches $(\mathbf{x}_i,Y_i)_{i=1..b}$ of size $b = 32$ running in parallel rather than a single random example, which reduces the variance in the rank estimate and results in better approximations to the real gradient. We also experimented with a "soft"-rank estimate $\mathrm{L}_r\left(\frac{m-|Y_i|}{\hat{n}}(1-\rho(\mathbf{y}_i,\hat{\mathbf{y}}))\right)$ taking into account the initial label correlation $\rho(\mathbf{y},\hat{\mathbf{y}}) \in [-1,..,1]$ in the annotation of the training data. However, on our benchmarks we did not observe a significant improvement in the retrieval performance.

4. Cross-Modal Embedding and Retrieval

Our supervised learning so far considered only labeled 3D-shapes. We now extend the method to handle images and line drawings. After that, we describe how we can learn a joint space of multi-modal data objects.

4.1. Image Data

For handling images, we use the same algorithmic framework. We only have to adapt the descriptors. For photographs, we use standard histogram of oriented gradient (HOG) features as introduced by Dalal and Triggs [DT05]. We use 6×6 HOG cells with 16 orientational bins, operating on gray-scale gradients. We compute dense descriptors sliding a window on the regular pixel grid and build bags of features as discussed before, with a dictionary size of 512 k-means clusters. In addition, we use a global, tiny 5×5 pixel color image in order to capture color and rough image layout. We employ these descriptors for photographs from the Labelme dataset [RES*06].

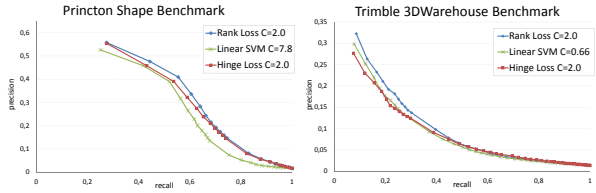
For line drawings (scanned 2D scribbles), we use the same principle setup with slightly different parameters ($4 \times 4 \times 8$) and dropping the color image.

4.2. Multi-Modal Embedding

The latent-space embedding makes it possible to learn the corresponding object mapping $\mathbf{V}$ (Section 3.2.2) jointly for different modalities. For example we are able to learn a mapping $\mathbf{V}_{img}$ for images and $\mathbf{V}_{3D}$ for 3D-shapes simultaneously where we use the common annotation as the glue to ensure that images and 3D-shapes that are semantically similar are also mapped in a similar direction in the shared latent space. Interestingly, this scheme not only allows for using completely different descriptors $\mathbf{x}$ for each modality, but benefits from sharing semantic labels across modalities, which, due to correlation with other common labels, are automatically transferred. For example, the labels *green* and *leaves* that we find as best matches for a 3D-shape query of a tree, do not exist in our Trimble 3D Warehouse dataset, but only in the image-based Labelme dataset. The only change to the learning procedure described in Section 3.2.2 required is to randomly alternate between the different modalities at each SGD step of the optimization. Interactive cross-modal retrieval results can be seen in the accompanying video. An extension to other modalities such as text-documents, sound, etc., is easily possible (relying only on suitable descriptors); this is left for future work.

Dataset	Type	Size	$\lvert Y\rvert$	$\lvert Y_i\rvert$	Descriptor
Princeton Shape Bench. [SMKF04]	3D meshes	907+907	114	2.4	BoW(512): HON(4,4,8)
Trimble 3D Warehouse™	3D meshes	1155+784	308	6	BoW(512): HOC(4,4,8)
Labelme [RTMF07]	photos	18538 +18500	767	4.9	BoW(512): HOG(6,6,16) + Color(5,5)
Scribbles	line-drawing	374	24	1.8	BoW(256): HOG(4,4,8)

Table 1: *Datasets used in our evaluation: Type, size (number of training + test objects), total number of label categories $\lvert Y\rvert$, average number of labels per object $\lvert Y_i\rvert$, feature descriptor (parameters: number of words; number of spatial and rotational bins; color has no rotational binning).*



(a) (b)

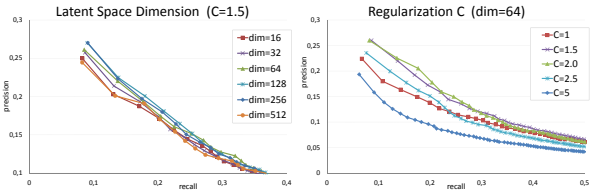
Figure 7: *Precision-recall curves for PSB and WH datasets. Methods are: our latent-space rank-loss model (Rank Loss) described in Section 3.2.2, one-versus-all linear SVM [FCH*08] with optimized hyper-parameter C (Linear SVM), and a linear SVM however with a latent space embedding (Hinge Loss). The latent space dimension is 128.*

5. Results

We evaluate the classification and retrieval performance of our method on different benchmarks (statistics are shown in Table 1 and precision-recall curves are shown in Figs. 7,8,9). Specifically, we have tested on the Princeton 3D-Shape Benchmark (PSB) [SMKF04], on a custom 3D-shape benchmark consisting of 1939 objects obtained by searching the Trimble 3D Warehouse™ (WH) for 34 different keywords. For our cross-modal retrieval, we use the popular *Labelme* image dataset [RTMF07]. It contains densely labeled high-resolution color-images and provides the shape and location of semantic objects within the images that we ignore in our evaluation. The *scribbles* dataset is a small collection of line-drawings (black and white images) that we have drawn and annotated ourselves for the purpose of demonstration (Fig. 12). For a quantitative assessment, we compare against the benchmark of Eitz et al. [ERB*12].

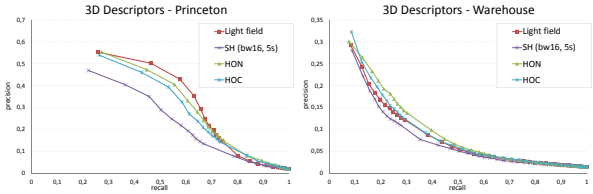
5.1. Relating Shapes

Base-line methods: We first compare our approach to common baseline classification tools. We begin with linear support vector machines (Linear SVM) [FCH*08], trained in a discriminative one-versus-all setting, thereby not utilizing feature sharing via the common latent space. For the experiment, we optimize the hyper-parameter C of the baseline SVMs using a grid-search with cross-validation. We also validate the proposed design by simplifying the individual steps: In particular, we test our log-based rank-loss (Rank Loss) in Eq. (11) against a max-margin L_1-loss (Hinge Loss), which ignores the rank and is conceptually similar to traditional collaborative filtering techniques for multi-task learning used in recommendation systems [RS05,LF08] rather than [WBH11]. The quantitative results (precision-recall curves) computed for the proposed HOC-descriptor (Sec. 3.1) are shown in Fig. 7 for the two 3D-shape retrieval benchmarks, PSB and WH. Since the PSB contains relatively "clean" 3D-meshes with hierarchically organized labels (e.g., *furniture* $\rightarrow$ *seat* $\rightarrow$ *chair*) the methods using feature sharing clearly outperform the one-versus-all SVMs in Fig. 7(a). However, using the rank-optimized loss only marginally improves the performance compared to the simpler hinge-loss. In particular, the top-prediction (precision@1) is nearly the same for the rank-loss and the hinge-loss. In contrast, the quality of the WH dataset varies strongly, making it difficult for multi-class classification methods. Here, due to noise and outliers the effect of robustly optimizing the top-ranked objects is very important



(a) (b)

Figure 8: *The effect of the 2 configurable parameters on the precision-recall for the Trimble 3D Warehouse™ Benchmark.*



(a) (b)

Figure 9: *The discriminative power of various 3D descriptors tested on (a) the Princeton shape benchmark (PSB) and (b) our Trimble 3D Warehouse™ benchmark.*

and clearly visible in Fig. 7(b): for small recall the test precision of our model trained with a log-rank loss is relatively high compared to the baseline methods.

Computational efficiency: One interesting property of our algorithm is its scalability with respect to data base and annotation size. Due to the rank-loss and the low-dimensional embedding our learning algorithm can effectively cope with many labels where the performance of common multi-class (e.g., one-versus-all) classifiers deteriorates quickly. Further, the run-time complexity at test time is lower than for linear one-versus-all classifiers since the latent space dimension can be lower than the total number of labels, leading to relatively few matrix-vector multiplications. In our experiments, we obtain interactive query performance on a desktop PC (see video). The training time is in the order of a few minutes for smaller datasets (Princeton or Trimble 3D Warehouse) but may take up to one hour for large datasets with many labels (e.g., Labelme), which is comparable to other large-scale learning algorithms such as linear SVMs [FCH*08]. However, the core algorithm is very easy to implement and, in addition, requires little memory in contrast to k-nearest neighbor classifiers and one-versus-all SVMs, in particular kernel SVMs.

Parameters: Our model needs the following parameters: the latent space dimension d, the regularization parameter C, the learning rate τ of the stochastic gradient descent (SGD). Like for standard SVMs, the most sensitive parameter is the regularization parameter C, which controls the amount of fitting of the model to the training data (the greater the higher the risk of overfitting). The influence of C on the classification performance for the Trimble 3D Warehouse test-dataset is shown in Fig. 8(b). As a rule of thumb, a value of $C = 2$ works well in our experiments. The latent space dimension parameter d is less critical than C and values of $d \in [100,..,200]$ give near optimal results for our datasets as shown in Fig. 8(a). For $d > 250$ test performance starts to drop since we lose the advantage of feature sharing – an important property for good generalization performance. However, further experiments with *ensemble models* that linearly combine several trained linear models of different dimensions [WBU11] may still give a significant performance boost. As in [WBU11] the learning rate τ of the stochastic gradient descent was set empirically – we used a constant value of $\tau = 0.1$ in all our tests. Experiments with data-adaptive learning rates and a decaying rate $\tau \propto 1/t$ did not improve our results but yielded much slower convergence.

Descriptors: We compare different 3D shape descriptors (see Sec. 3.1) with respect to their prediction performance within our algorithm. Particularly, we test the rotation-invariant spherical harmonics descriptor (SH) [KFR03], the light-field descriptor (LF) [CXPYTO03], and our HOC and HON descriptors computed at different scales that are similar to the MeshHOG descriptor [ZBVH09]. Results are shown in Fig. 9. For all descriptors we use a bag-of-feature approach to generate a final single histogram from the locally computed descriptors for each 3D model. Local LF descriptors are generated by densely sampling view points on the model's bounding sphere. The SH descriptors are computed at bandwidth $b = 16$ for 5 concentric spherical shells with centers sampled uniformly across the 3D model. The HOC and HON descriptors are computed as described in Sec. 3.1. LF descriptors show high precision values on the PSB dataset, due to the discriminative silhouette renderings obtainable from the PSB models. Because of many low-res., locally planar 3D models HON descriptors perform slightly better than HOC descriptors, while SH descriptors yield much lower precision overall – perhaps due to the lost orientational information. On the WH dataset we found that HOC and HON descriptors give the best results. HOC descriptors yield a better precision for the top label, whereas HON descriptors perform slightly better for the lower-rank predictions. Due to many complex models in this dataset, the silhouette-based LF descriptors perform slightly worse.

5.2. Cross-Modal Embedding

As a proof-of-concept we also evaluated our cross-modal retrieval performance on a publicly available benchmark for sketch-based 3D shape retrieval [ERB*12]. This benchmark extends the Princeton Shape Benchmark (PSB) with a corresponding human-scribbled 2D-image sketch for each 3D shape. Then, instead of text labels as queries, the sketches are used to retrieve the best matching 3D shapes. For running the benchmark we precompute HON descriptors for all 3D shapes and HOG descriptors for the 2D scribble images

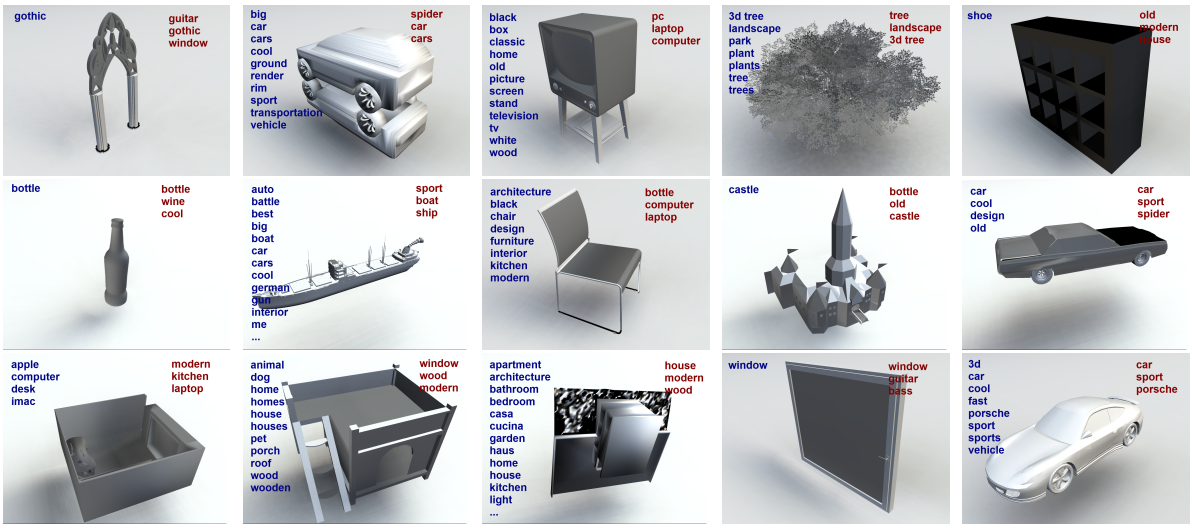
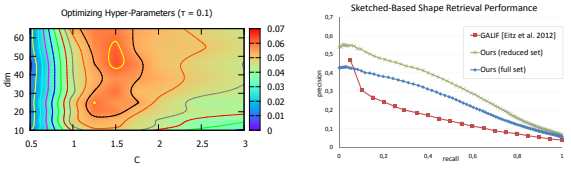


Figure 10: *This figure shows an excerpt of the results for automatic annotation of the Trimble 3D Warehouse test set using our rank-based learning algorithm that was trained on 3D HOC bag-of-feature descriptors as described in Section 3.2.1. The 3 top ranked predictions out of 308 label categories are shown in red in the top-right corner (top-left shows the ground-truth).*

with parameters denoted in Table 1. Following [ERB*12] we also optimize the hyper-parameters on the training set of this benchmark using a grid search with cross-validation, which is shown in Fig. 11a. However, in contrast to [ERB*12] we do not optimize the hyper-parameters of the descriptors but rather optimize the cross-modal embedding parameters jointly for scribble and 3D shape descriptors given the common annotations in the PSB training data (see Sec. 4.2). Since this benchmark was not prepared for learning a classifier/distance metric, approximately half of the labels (68 of 131) found in the test set do not appear in the training set and therefore, can not be learned. By excluding the instances "unseen" in the training set from the original benchmark we obtain the retrieval performance shown by the green curve in Fig. 11b. In addition, we modify the benchmark such that for each label category in the test set there is at least one sample found in the training set, which results in the blue curve in Fig. 11b. Please note that the performance of the original method [ERB*12] (red curve) is therefore not directly comparable with our results and only plotted as a baseline.



(a) optimizing hyper-parameters (b) precision/recall

Figure 11: *Cross-modal retrieval performance with scribble queries for the PSB dataset [SMKF04], based on the benchmark from [ERB*12]. (a) The color-coded contour plot shows the area-under-curve (auc) of the precision-recall curve for the top 20 retrieved meshes with varying regularization parameter C and latent space dimension (dim). The optimal setting is found at $C = 1.5$, $dim = 46$, which we used to compute the precision-recall curves in (b).*

5.3. Applications

We have applied our method to two scenarios: label suggestion for sparsely annotated and noisy shape data bases like Trimble 3D Warehouse, and cross-modal object retrieval and data exploration.

Automatic Annotations for 3D-Object Databases

Since predictions on the PSB dataset are relatively good, we show only results for 3D-object annotation on the noisy Trimble 3D Warehouse dataset in Fig. 10. Due to noisy labels and the lack of explicit context modeling in our bag-of-feature descriptor, some results are wrongly labeled. Nonetheless, in most cases at least one of the top three predicted labels recognizes the object category correctly. In cases where this does not apply such as the model of a TV set (middle image in the upper row of Fig. 10), the suggestions are still plausible (the TV set is recognized as “computer”).

Given that the 3D Warehouse data is very noisy and the method works fully unsupervised (given the downloaded data), the results are already encouraging.

Cross Modal Retrieval

In Fig. 12, we show an example of the retrieval results for

joint cross-modal learning of images (annotated photographs from Labelme), 3D-shapes (Trimble 3D Warehouse), and user scribbles (custom dataset). We have implemented a simple user interface that permits an interactive navigation of the latent space (showing the current query in the middle and the nearest neighbors in concentric arrangement; the size of the results reflects the rank). Please also refer to the accompanying video for an interactive demo. We also provide further screenshots of interactive sessions in the supplemental material. The nearest neighbor sets are predominantly matching the category of the query objects but some mismatches show up. For interactive browsing, this is no limitation. We can also see how the results show objects that are related in shape, appearance, or textual labeling, which gives us a nice way of exploring the data space.

6. Conclusions and Future Work

We have presented a system for learning the assignment of labels to 3D geometry by learning a low-rank classifier matrix that recognizes similarities of labels through correlations in shape. This permits information sharing across geometrically similar objects as well as semantically related labels. In experiments, we can clearly see an advantage in performance over baseline methods that ignore this side information. Moreover, we have generalized the idea of multi-label classification through a low-dimensional latent space to obtain a novel cross-modal embedding of objects. It can be used for object retrieval across different modalities and for interactive explorations of complex data spaces. To this end, we also evaluated the performance of several 3D shape descriptors for object retrieval when using a discriminatively trained similarity metric.

In future work, we would like to extend our model to also learn to localize 3D-shapes within larger 3D scenes, which is particularly important for annotating and recognizing objects in 3D point clouds. Similar to object detection in images, this requires searching for "high-scoring" parts using a sliding-window approach or employing a constellation model that explicitly learns the context of 3D-objects.

Acknowledgements

The research for this work was partially funded by the support of grants NSF DMS 1228304, AFOSR FA9550-12-1-0372, ONR MURI N00014-13-1-0341, a Google research award, and the Max Planck Center for Visual Computing and Communication. The authors would like to thank Bernt Schiele and Martin Bokeloh for discussions on the descriptor design.

References

[ARS13] AUBRY M., RUSSELL B., SIVIC J.: Painting-to-3D model alignment via discriminative visual elements. *ACM Transactions on Graphics* (2013). 1, 3

[ATC*05] ANGUELOV D., TASKAR B., CHATALBASHEV V., KOLLER D., GUPTA D., HEITZ G., NG A.: Discriminative learning of markov random fields for segmentation of 3d scan data. In *CVPR* (2005). 1

[BB08] BOTTOU L., BOUSQUET O.: The tradeoffs of large scale learning. *Adv. in Neural Information Processing Systems 20* (2008). 6

[BBGO11] BRONSTEIN A. M., BRONSTEIN M. M., GUIBAS L. J., OVSJANIKOV M.: Shape Google: Geometric words and expressions for invariant shape retrieval. *ACM Trans. Graph. 30* (February 2011). 2

[CB04] COLLOBERT R., BENGIO S.: Links between perceptrons, mlps and svms. *Proceedings of ICML* (2004). 3, 6

[CP05] CAZALS F., POUGET M.: Estimating differential quantities using polynomial fitting of osculating jets. *Computer Aided Geometric Design 22*, 2 (2005), 121 – 146. 3

[CXPYTO03] CHEN D.-Y., X.-P.TIAN, Y.-T.SHEN, OUHYOUNG M.: On visual similarity based 3d model retrieval. *In Proc. of Eurographics 22*, 3 (2003), 223 – 232. 3, 8

[DT05] DALAL N., TRIGGS B.: Histograms of oriented gradients for human detection. In *Proc. Conf. Comp. Vision and Pattern Recognition CVPR* (2005), pp. 886 – 893. 2, 6

[ERB*12] EITZ M., RICHTER R., BOUBEKEUR T., HILDEBRAND K., ALEXA M.: Sketch-based shape retrieval. *ACM Trans. Graph. (Proc. SIGGRAPH) 31*, 4 (2012), 1–10. 2, 3, 7, 8, 9

[FCH*08] FAN R.-E., CHANG K.-W., HSIEH C.-J., WANG X.-R., LIN C.-J.: Liblinear: A library for large linear classification. *Journal of Machine Learning Research* (2008). 2, 7, 8

[FEH10] FARHADI A., ENDRES I., HOIEM D.: Attribute-centric recognition for cross-category generalization. In *CVPR* (2010), pp. 2352–2359. 3

[FEHF09] FARHADI A., ENDRES I., HOIEM D., FORSYTH D.: Describing objects by their attributes. In *Proc. Conf. Comp. Vision and Pattern Recognition CVPR* (2009), pp. 1778–1785. 3

[FGMR10] FELZENSZWALB P., GIRSHICK R., MCALLESTER D., RAMANAN D.: Object detection with discriminatively trained part-based models. *IEEE Trans. on Pattern Analysis and Machine Intelligence 32*, 9 (2010), 1627–1645. 6

[HFL14] HUANG Z., FU H., LAU R. W. H.: Data-driven segmentation and labeling of freehand sketches. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2014)* (2014). 1

[HSS*13] HUANG S.-S., SHAMIR A., SHEN C.-H., ZHANG H., SHEFFER A., HU S.-M., COHEN-OR D.: Qualitative organization of collections of shapes via quartet analysis. *ACM Transactions on Graphics 32*, 4 (2013), 1–10. 1, 3

[KBWS13] KERBER J., BOKELOH M., WAND M., SEIDEL H.-P.: Scalable symmetry detection for urban scenes. *Computer Graphics Forum* (2013), 3–15. 2, 3

[KFLCO13] KLEIMAN Y., FISH N., LANIR J., COHEN-OR D.: Dynamic maps for exploring and browsing shapes. *Computer Graphics Forum (Proceedings of SGP)* (2013). 1, 3

[KFR03] KAZHDAN M., FUNKHOUSER T., RUSINKIEWICZ S.: Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Symposium on Geometry Processing* (2003). 3, 8

[KHS10] KALOGERAKIS E., HERTZMANN A., SINGH K.: Learning 3d mesh segmentation and labeling. *ACM Trans. Graph. 29*, 3 (2010). 1

[LBBC14] LITMAN R., BRONSTEIN A., BRONSTEIN M., CASTELLANI U.: Supervised learning of bag-of-features shape

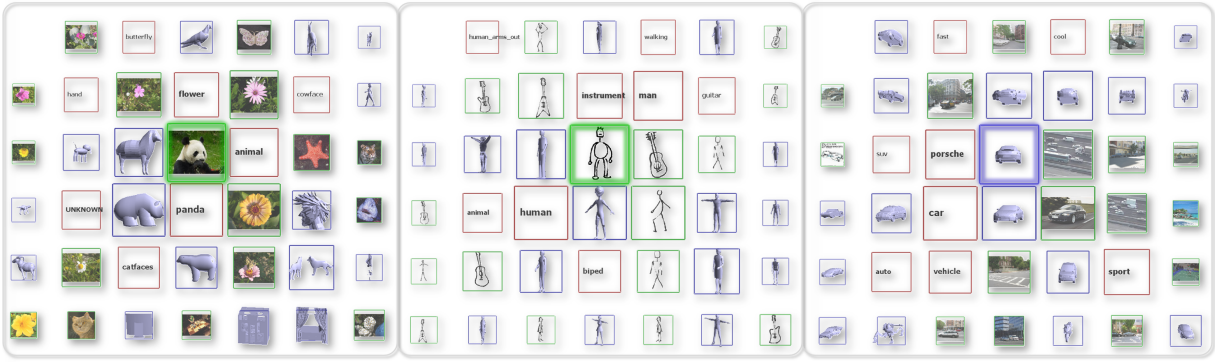


Figure 12: *Screenshots of our application for cross-modal shape retrieval. The user can interactively select a 3D-shape, image, type a text query, or simply scribble a sketch and retrieves the highest ranked results from all modalities. Here, the query data base consists of 3 modalities: images (Labelme dataset [RTMF07]), 3D-shapes from Trimble 3D Warehouse, and text labels (union of datasets); or labels, 3D-shapes (Princeton Shape Benchmark [SMKF04]) and scribbles (center). Queries are highlighted in the center and results are arranged as: the closer the result to the center the higher its rank (size of the bounding box reflects its relevance). For more examples browse the supplemental material and watch the accompanying video material.*

descriptors using sparse coding. *Computer Graphics Forum (Proceedings of SGP)* (2014). 2

[LF08] LOEFF N., FARHADI A.: Scene discovery by matrix factorization. In *Proc. of ECCV* (2008), pp. 451–464. 2, 3, 5, 7

[LFEF09] LOEFF N., FARHADI A., ENDRES I., FORSYTH D. A.: Unlabeled Data Improves Word Prediction. In *Int'l Conf. on Computer Vision* (2009). 2, 3, 5

[LG07] LI X., GUSKOV I.: 3d object recognition from range images using pyramid matching. In *ICCV, Workshop on 3D Representation for Recognition* (2007). 1, 2

[Low04] LOWE D.: Distinctive image features from scale-invariant keypoints. *IJCV 20* (2004), 91–110. 2

[LSP06] LAZEBNIK S., SCHMID C., PONCE J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR* (2006). 2

[MFK*10] MAES C., FABRY T., KEUSTERMANS J., SMEETS D., SUETENS P., VANDERMEULEN D.: Feature detection on 3d face surfaces for pose normalisation and recognition. In *BTAS* (2010). 2

[MGGP06] MITRA N. J., GUIBAS L., GIESEN J., PAULY M.: Probabilistic fingerprints for shapes. In *Symposium on Geometry Processing* (2006), pp. 121–130. 1, 2

[RES*06] RUSSELL B. C., EFROS A. A., SIVIC J., FREEMAN W. T., ZISSERMAN A.: Using multiple segmentations to discover objects and their extent in image collections. *CVPR* (2006). 1, 6

[RMHM14] RAHMANI H., MAHMOOD A., HUYNH D. Q., MIAN A.: Hopc: Histogram of oriented principal components of 3d pointclouds for action recognition. In *ECCV* (2014). 2, 3

[RS05] RENNIE J. D. M., SREBRO N.: Fast maximum margin matrix factorization for collaborative prediction. *Proceedings of ICML* (2005), 713–719. 2, 3, 5, 7

[RTMF07] RUSSELL B., TORRALBA A., MURPHY K., FREEMAN W. T.: Labelme: a database and web-based tool for image annotation. *Int'l Journal of Computer Vision* (2007). 2, 7, 11

[SK09] SU X., KHOSHGOFTAAR T. M.: A survey of collaborative filtering techniques. *Adv. in Artif. Intell.* (2009). 2, 3

[SKVS13] SMEETS D., KEUSTERMANS J., VANDERMEULEN D., SUETENS P.: MeshSIFT: Local surface features for 3d face recognition under expression variations and partial data. *Comp. Vision and Img. Understanding 117*, 2 (2013), 158 – 169. 2, 3

[SMKF04] SHILANE P., MIN P., KAZHDAN M., FUNKHOUSER T.: The princeton shape benchmark. *Proc. Shape Modeling International* (2004), 167–168. 1, 4, 7, 9, 11

[ST14] SCHNEIDER R. G., TUYTELAARS T.: Sketch classification and classification-driven analysis using fisher vectors. *ACM Trans. Graph. 33*, 6 (Nov. 2014), 174:1–174:9. 3

[STS14] SALTI S., TOMBARI F., STEFANO L. D.: Shot: Unique signatures of histograms for surface and texture description. *Comp. Vis. and Img. Understanding* (2014), 251 – 264. 2, 3

[SX14] SONG S., XIAO J.: Sliding shapes for 3d object detection in depth images. In *ECCV* (2014). 2, 3

[TV08] TANGELDER J., VELTKAMP R.: A survey of content based 3d shape retrieval methods. *Multimedia Tools and Applications 39* (2008), 441–471. 2

[vKZHCO11] VAN KAICK O., ZHANG H., HAMARNEH G., COHEN-OR D.: A survey on shape correspondence. *Computer Graphics Forum 30*, 6 (2011), 1681–1707. 1, 2

[WBH11] WESTON J., BENGIO S., HAMEL P.: Large-scale music annotation and retrieval: Learning to rank in joint semantic spaces. *Journal of New Music Research* (2011). 3, 7

[WBU11] WESTON J., BENGIO S., USUNIER N.: Wsabie: scaling up to large vocabulary image annotation. In *Proceedings of IJCAI* (2011), pp. 2764–2770. 1, 2, 3, 5, 6, 8

[WdW10] WIJNHOVEN R. G., DE WITH P. H.: Fast training of object detection using stochastic gradient descent. *Int'l Conference on Pattern Recognition* (2010). 6

[XMZ*14] XU K., MA R., ZHANG H., ZHU C., SHAMIR A., COHEN-OR D., HUANG H.: Organizing heterogeneous scene collection through contextual focal points. *ACM Transactions on Graphics, (Proc. of SIGGRAPH 2014) 33*, 4 (2014). 3

[ZBVH09] ZAHARESCU A., BOYER E., VARANASI K., HORAUD R.: Surface Feature Detection and Description with Applications to Mesh Matching. *CVPR* (2009), 373–380. 3, 8